Algoritmos y Estructuras de Datos II
Primer parcial — Viernes 16 de Septiembre de 2016

Aclaraciones

- El parcial es **a libro abierto**.
- Cada ejercicio debe entregarse **en hojas separadas**.
- Incluir en cada hoja el número de orden asignado, número de hoja, apellido y nombre.
- Al entregar el parcial, completar el resto de las columnas en la planilla.
- Cada ejercicio se calificará con **P**romocionado, **A**probado, **R**egular, o **I**nsuficiente.
- El parcial completo está aprobado si el primer ejercicio tiene al menos **A**, y entre los ejercicios 2 y 3 hay al menos una **A**. Para más detalles, ver "Información sobre la cursada" en el sitio Web.

Ej. 1. Especificación

En la red social "Roberto Carlos", cada usuario quiere tener un millón de amigos. Pueden registrarse nuevos usuarios a la red social en todo momento. Cuando lo deseen, dos usuarios pueden volverse amigos; la amistad es siempre recíproca, pero no es reflexiva. Las personas de esta red son muy serias al momento de declararse amigos: no hay arrepentimientos.

Con el objetivo de que una persona agrande su círculo de amistades, la red le sugiere a cada persona un usuario **no amigo** que posea la mayor cantidad de amigos en común con él.

También a la red social le interesa que los amigos se comuniquen. Por lo tanto, la red crea automáticamente para cada persona un grupo con sus amigos y con los amigos de éstos, para que esas personas compartan sus anécdotas. Si otro usuario se hace amigo de él, instantáneamente empieza a formar parte del grupo (y de los grupos de sus amigos).

Para que los usuarios tímidos utilicen más el sistema, la red social posee un conjunto de usuarios automatizados denominados "e-ameos" definidos de antemano. Cada vez que un usuario quiere hablar con alguien pero no quiere socializar, puede hacerse amigo de uno o más de los e-ameos de la red social. Sin embargo, la red social **no agregará** al grupo del usuario los amigos de los e-ameos y tampoco los ofrecerá como sugerencia de amistad a los usuarios ya que son públicamente conocidos.

Además, los diseñadores de la red notaron que, estadísticamente, cuando dos personas tienen 10 e-ameos en común, la probabilidad de que éstos se hagan amigos es del 99.9 %. Por lo tanto, la red les simplifica la vida *amigándolos automáticamente al* darse esta situación.

Modelar con un TAD la red social Roberto Carlos. Además, con fines estadísticos, interesa saber:

a) qué grupos existen y con qué integrantes;
b) el usuario más "figureti", es decir, el usuario no e-ameo que está en más grupos de amigos distintos de la red (en caso de existir empate, devolver alguno de manera determinista);
c) cuántos usuarios "Forever Alone" hay, es decir, usuarios que estén en grupo solamente con e-ameos.

Ej. 2. Inducción estructural

Dado el siguiente TAD con sus operaciones:

TAD ÁRBOL BIN-TERN(α)

géneros abt(α)

generadores

$\text{nil} : \longrightarrow \text{abt}(\alpha)$
$\text{bin} : \text{abt}(\alpha) \times \text{abt}(\alpha) \times \alpha \longrightarrow \text{abt}(\alpha)$
$\text{tern} : \text{abt}(\alpha) \times \text{abt}(\alpha) \times \text{abt}(\alpha) \times \alpha \longrightarrow \text{abt}(\alpha)$

otras operaciones

$\text{h} : \text{abt}(\alpha) \longrightarrow \text{nat}$
$\text{tam} : \text{abt}(\alpha) \longrightarrow \text{nat}$
$\text{completo} : \text{abt}(\alpha) \longrightarrow \text{bool}$

observadores básicos

$\text{nil?} : \text{abt}(\alpha) \longrightarrow \text{bool}$
$\text{tern?} : \text{abt}(\alpha) \longrightarrow \text{bool}$
$\text{raiz} : \text{abt}(\alpha)\ a \longrightarrow \alpha \quad \{\neg\text{nil?}(a)\}$
$\text{hijo}_1 : \text{abt}(\alpha)\ a \longrightarrow \text{abt}(\alpha) \quad \{\neg\text{nil?}(a)\}$
$\text{hijo}_2 : \text{abt}(\alpha)\ a \longrightarrow \text{abt}(\alpha) \quad \{\neg\text{nil?}(a)\}$
$\text{hijo}_3 : \text{abt}(\alpha)\ a \longrightarrow \text{abt}(\alpha) \quad \{\text{tern?}(a)\}$

Fin TAD

axiomas

$h_0)\ \text{h}(\text{nil}) \equiv 0$
$h_1)\ \text{h}(\text{bin}(i,d,e)) \equiv 1 + \max(\text{h}(i), \text{h}(d))$
$h_2)\ \text{h}(\text{tern}(a,b,c,e)) \equiv 1 + \max_3(\text{h}(a), \text{h}(b), \text{h}(c))$
$t_0)\ \text{tam}(\text{nil}) \equiv 0$
$t_1)\ \text{tam}(\text{bin}(i,d,e)) \equiv 1+\text{tam}(i)+\text{tam}(d)$
$t_2)\ \text{tam}(\text{tern}(a,b,c,e)) \equiv 1+\text{tam}(a)+\text{tam}(b)+\text{tam}(c)$
$c)\ \text{completo}(a) \equiv \text{nil?}(a) \lor_L \text{h}(\text{hijo}_1(a)) = \text{h}(\text{hijo}_2(a)) \land \text{completo}(\text{hijo}_1(a)) \land \text{completo}(\text{hijo}_2(a)) \land (\text{tern?}(a) \Rightarrow_L (\text{h}(\text{hijo}_2(a)) = \text{h}(\text{hijo}_3(a)) \land \text{completo}(\text{hijo}_3(a))))$

Demuestre por inducción estructural la siguiente propiedad:

$$\boxed{(\forall a : \text{abt}(\alpha))\ \text{completo}(a) \Rightarrow \text{tam}(a) \geq 2^{h(a)} - 1\ )}$$

En caso de utilizar lemas auxiliares, plantearlos claramente y demostrarlos. Además, se pide:

a) Escribir el predicado unario. Luego escribir, completo, **el esquema de inducción** a utilizar. En el esquema, marcar **claramente** CB(s), PI(s), HI(s), TI(s) y el alcance de cada cuantificador.

b) Plantear el/los caso(s) base y resolverlo(s), justificando cada paso de la demostración.

c) Plantear el/los paso(s) inductivo(s) y resolverlo(s), justificando cada paso de la demostración.

Ej. 3. Diseño

Considerar la siguiente especificación de un sistema que realiza el seguimiento de experimentos en un laboratorio. Los experimentos se enumeran por números naturales desde el 1 en adelante. De cada experimento sólo se recuerdan los reactivos químicos utilizados y el orden en el que fueron usados (el "protocolo"). Un mismo reactivo puede ser usado múltiples veces en el mismo protocolo. Los reactivos se identifican por su número de inventario que es un NAT, y cada uno tiene asociado un índice de peligrosidad representado por un NAT. Por cuestiones de seguridad, la peligrosidad combinada de todos los reactivos usados en un protocolo no puede superar el valor 100.

TADs REACTIVO, PELIGROSIDAD y ID son NAT

TAD PROTOCOLO es SECUENCIA(TUPLA(REACTIVO,PELIGROSIDAD))

TAD EXPERIMENTOS

observadores básicos

cantExperimentos : experimentos ⟶ nat

verExperimento : experimentos e × id i ⟶ protocolo {$1 \leq i \leq$ cantExperimentos(e)}

generadores

abrirLaboratorio : ⟶ experimentos

registrarExperimento : experimentos e × protocolo p ⟶ experimentos

$\left\{ \begin{array}{l} \neg\ \text{vacía?}(p)\ \wedge\ \sum_{i=1}^{long(p)} \Pi_2(p[i]) \leq 100\ \wedge \\ (\forall\ n : \text{id})\ n \leq \text{cantExperimentos}(e) \Rightarrow_L \text{peligrosidadesConsistentes}(\text{verExperimento}(e,n)\ \&\ p) \end{array} \right\}$

axiomas

cantExperimentos(abrirLaboratorio) ≡ 0

cantExperimentos(registrarExperimento(e, s)) ≡ 1+cantExperimentos(e)

verExperimento(i,registrarExperimento(e, s))) ≡ **if** i=cantExperimentos(e) **then** s **else** verExperimento(i, e) **fi**

predicados

peligrosidadesConsistentes(p) ≡ $(\forall i, j : nat)\ (1 \leq i, j \leq long(p) \Rightarrow_L (\Pi_1(p[i]) = \Pi_1(p[j]) \Rightarrow_L \Pi_2(p[i]) = \Pi_2(p[j])))$

Fin TAD

Se decidió utilizar la siguiente estructura para representar el TAD, en la que se pretende abreviar partes de los protocolos repetidas para reducir el espacio ustilizado para almacenarlas.

Experimentos **se representa con** estr

donde estr es tupla ⟨ *cantUsos*: dicc(reactivo,conj(id)),
porPeligrosidad: dicc(peligrosidad,conj(reactivo)),
protocolos: dicc(id, secu(nombre)),
subProtocolos: dicc(nombre, secu(reactivo)) ⟩

y nombre es STRING

En esta estructura, *cantUsos* almacena los números de experimento en los que se usó cada reactivo; *porPeligrosidad* clasifica a los reactivos según su peligrosidad (sólo hay entradas para peligrosidades usadas, i.e., no hay definiciones que sean el conjunto vacio). Además, *protocolos* describe abreviadamente el protocolo usado en cada experimento como una secuencia de nombres de subprotocolos (concatenando esos subprotocolos se obtiene el protocolo completo), y *subProtocolos* almacena los pedazos de secuencias de reactivos correspondientes a cada subprotocolo *usado en algún protocolo existente*. De esta manera, el mismo subprotocolo se puede reutilizar para describir múltiples protocolos.

Se pide:

a) Escribir en castellano el invariante de representación.

b) Escribir formalmente el invariante de representación.

c) Escribir formalmente la función de abstracción.

Teodoro Freund 526/15 P

HOJA N° 1

ORDEN 12

① TAD usuario ES NAT | 1 P | 2 P | 3 P | FRANCISCO S.

TAD RC

igualdad observacional

$(\forall a, b : rc) \left(a =_{obs} b \iff \begin{pmatrix} Usuarios(a) = Usuarios(b) \land \\ Ameos(a) = Ameos(b) \land_L \\ (\forall u : usuario)\ u \in (Usuarios(a) - Ameos(a)) \Rightarrow_L \\ Amigos(u, a) = Amigos(u, b) \end{pmatrix} \right)$

exporta

usa conj

géneros rc

Observadores básicos

- Usuarios : rc ⟶ conj(usuario)
- Ameos : rc ⟶ conj(usuario)
- Amigos : usuario u × rc r ⟶ conj(usuario)

~~{u ∈ Usuarios(r)}~~ {u ∈ Personas(r)}

generadores

iniciar : conj(usuario) ⟶ rc

regusuario : usuario u × rc r ⟶ rc

{u ∉ Usuarios(r)}

amigar : usuario U_1 × usuario U_2 × rc r ⟶ rc

$\{U_1 \neq U_2 \land \{U_1, U_2\} \subseteq Usuarios(r) \land (U_1 \in Personas(r) \lor U_2 \in Personas(r))\}$

NOTA

otras operaciones

Grupo : usuario u × rc r ⟶ conj(usuario)

$\{u \in \text{Personas}(r)\}$

· GrupoAux : ~~conj(usuario)~~ × conj(usuario) c_2 × rc r ⟶ conj(usuario)

$\{c_2 \subseteq \text{Usuarios}(r)\}$

Personas : rc ⟶ conj(usuario)

· AmigosAux : conj(usuario) c_1 × conj(usuario) c_2 × rc r ⟶ conj(usuario)

$\{c_1 \subseteq \text{Usuarios}(r) \wedge c_2 \subseteq \text{Personas}(r)\}$

· Sugerencia : usuario u × rc r ⟶ usuario

$\{u \in \text{Personas}(r) \wedge (\exists u' : \text{usuario})(u' \in \text{Personas}(rc) \wedge u' \notin \text{Amigos}(u, rc$

– SugAux : conj(usuario) us_1 × nat × conj(usuario) us_2 × conj(usuario) us_3 × rc r ⟶ conj(usuario)

$\{us_1 \subseteq \text{Usuarios}(r) \wedge us_2 \cup us_3 \subseteq \text{Personas}(r)\}$

Figureti : rc ⟶ usuario falta restricción.

· FigAux : conj(usuario) c_1 × nat × conj(usuario) c_2 × rc r ⟶ conj(usuario)

$\{c_1 \subseteq \text{Personas}(r) \wedge c_2 \subseteq \text{Personas}(r)\}$

ForAlo : conj(usuario) c_1 × rc r ⟶ nat

$\{c_1 \subseteq \text{Personas}(r)\}$

Axiomas

Usuarios(iniciar(c)) $\equiv$ c

Usuarios(regusuario(u, rc)) $\equiv$ Ag(u, Usuarios(rc))

Usuarios(amigar(u_1, u_2, rc)) $\equiv$ Usuarios(rc)

Ameos(iniciar(c)) $\equiv$ c

Ameos(regusuario(u, rc)) $\equiv$ Ameos(rc)

Ameos(amigar(u_1, u_2, rc)) $\equiv$ Ameos(rc)

Amigos(u, iniciar(c)) $\equiv \emptyset$

Amigos(u, regusuario(u', rc)) $\equiv$ if u = u' then $\emptyset$ else
Amigos(u, rc) fi

Amigos(u, amigar(u_1, u_2, rc)) $\equiv$

if $\{u_1, u_2\} \subseteq$ Personas(rc) then
if $u \in \{u_1, u_2\}$ then Amigos(u, rc) $\cup$ ($\{u_1, u_2\} - u$)
else Amigos(u, rc) fi

⊗ imposible leer esto.

else if $u \in \{u_1, u_2\}$ then
AmigosAux(($\{u_1, u_2\} - u$) $\cup$ Amigos(u, rc)), Personas(rc), rc)
$\cup$ Amigos(u, rc)
else

Recordar: es una documentación. No un algoritmo.

if $u_1 \in$ Ameos(rc) then
if #(Amigos(u, rc) $\cap$ Ag(u_1, Amigos(u_2, rc)) $\cap$ Ameos(rc)) $\geq$ 10
then Ag(u_2, Amigos(u, rc))
else Amigos(u, rc) fi

```
            else if #(Amigos(u, rc) ∩ Ag(u2, Amigos(u1, rc))
                      ∩ Ameos(rc)) ≥ 10 then
                 Ag(u1, Amigos(u, rc))
              else Amigos(u, rc) fi fi fi fi

AmigosAux(a, p, rc) ≡
      if ∅?(p) then ∅ else
        if #(a ∩ Amigos(DameUno(p), rc) ∩ Ameos(rc)) ≥ 10
          then {DameUno(p)} else ∅ fi ∪ AmigosAux(a, SinUno(p), rc) fi

Personas(rc) ≡ Usuarios(rc) − Ameos(rc)
Grupo(u, rc) ≡ GrupoAux(Amigos(u, rc), rc)
GrupoAux(am, rc) ≡ if ∅?(am) then ∅ else
                 if DameUno(am) ∈ Ameos(rc) then ∅
                 else Amigos(DameUno(am), rc) fi ∪
                 Ag(DameUno(am), GrupoAux(sinUno(am), rc)) fi

Sugerencia(u, rc) ≡ DameUno(SugAux(Amigos(u, rc), 0, ∅, Personas(rc), rc))
SugAux(am, cnt, res, p, rc) ≡ if ∅?(p) then res else
                 if #(am ∩ Amigos(DameUno(p), rc)) > cnt then
                   SugAux(am, #(am ∩ Amigos(DameUno(p), rc)),
                          {DameUno(p)}, SinUno(p), rc)
                 else if #(am ∩ Amigos(DameUno(p), rc)) = cnt then
                   SugAux(am, cnt, {DameUno(p)} ∪ res, sinUno(p), rc)
                 else SugAux(am, cnt, res, sinUno(p), rc) fi fi fi
```

esto es un algoritmo,
no una espec.
sería mucho descriptivo
en castellano para ver.

↳ Hay que buscar ~~una~~ otra mejor forma de escribir un máximo.

$\to \perp$ cuando $Personas(rc) = \emptyset$. ORDEN 17
Falta restricción

$Figureti(rc) \equiv DameUno(FigAux(\emptyset, 0, Personas(rc), rc))$

$FigAux(res, cnt, p, rc) \equiv$ if $\emptyset?(p)$ then res else

if $\#(\cancel{Amigos}\ Grupo(DameUno(p), rc) \cap Personas(rc)) > cnt$ then

$FigAux(\{DameUno(p)\}, \#(\cancel{Amigos}\ Grupo(DameUno(p), rc)),$

¡Idem

$SinUno(p), rc)$

else if $\#(\cancel{Amigos}\ Grupo(DameUno(p), rc) \cap Personas(rc)) = cnt$ then

$FigAux(res \cup \{DameUno(p)\}, cnt, SinUno(p), rc)$

else $FigAux(res, cnt, SinUno(p), rc)$ fi fi fi

$ForAlo(ps, rc) \equiv$ if $\emptyset?(ps)$ then 0 else

if $Grupo(DameUno(ps), rc) \in Amenos(rc)$ then 1

else 0 fi + $ForAlo(SinUno(ps), rc)$ fi

Aclaraciones

- Notar que no consideramos a los amigos personas, no nos interesan sus amigos ni su grupo
- Además, cada persona ~~pertenece~~ a tantos grupos como ~~amigos~~ ~~tenga~~ miembros no amigos de su grupo tenga

② a. $P(a) : \text{completo}(a) \Rightarrow \text{tam}(a) \geq 2^{h(a)} - 1$

$$\overbrace{P(nil)}^{CB} \land \overbrace{(\forall a,b:abt(\alpha))\ \overbrace{P(a)\land P(b)}^{HI} \Rightarrow \overbrace{(\forall x:\alpha)\ P(bin(a,b,x))}^{TI}}^{PI}$$

$$\land \ (\forall a,b,c:abt(\alpha))\ \underbrace{\underbrace{P(a)\land P(b)\land P(c)}_{HI} \Rightarrow \underbrace{(\forall x:\alpha)\ P(tern(a,b,c,x))}_{TI}}_{PI}$$

b. Quiero Probar $P(nil) \Leftrightarrow \text{completo}(nil) \Rightarrow \text{tam}(nil) \geq 2^{h(nil)} - 1$

REEMPLAZO POR ~~a~~, t_0), h_0)

$$\text{completo}(nil) \Rightarrow \underbrace{0 \geq 2^0 - 1 = 1 - 1 = 0}_{\equiv\ true}$$

Luego, por ser true el consecuente de la implicación, y por no indefinirse completo, vale $P(nil)$

c. $(\forall a,b:abt(\alpha))\ P(a) \land P(b) \Rightarrow (\forall x:\alpha)\ P(bin(a,b,x))$

NO ASUMO NADA SOBRE a y b, Y SACO EL $\forall$

$$P(a) \land P(b) \Rightarrow (\forall x:\alpha)\ P(bin(a,b,x))$$

Si $\neg(P(a) \land P(b))$ LA DEMOSTRACIÓN ES TRIVIAL, LAS ASUMO VERDADERAS y continuo

$(\forall x:\alpha)\ P(bin(a,b,x))$ QUITO EL $\forall$ SIN ASUMIR NADA SOBRE x

$P(bin(a,b,x))$ QUE SE TRADUCE EN

$$\text{completo}(bin(a,b,x)) \Rightarrow \text{tam}(bin(a,b,x)) \geq 2^{h(bin(a,b,x))} - 1$$

POR EL AXIOMA c)

$nil?(bin(a,b,x)) \vee_L h(hijo_1(bin(a,b,x))) = h(hijo_2(bin(a,b,x))) \wedge completo(hijo_1(bin(a,b,x))) \wedge completo(hijo_2(bin(a,b,x))) \wedge (tern?(bin(a,b,x)) \Rightarrow_L (\ldots\ldots)) \Rightarrow tam(bin(a,b,x)) \geq 2^{h(bin(a,b,x))} - 1$

NOTAR QUE $nil?(bin(a,b,x)) \equiv false$

y $tern?(bin(a,b,x)) \equiv false$, ENTONCES, POR SER $\vee_L$ y un $\Rightarrow_L$

SOLO INTERESA ESTO, TAMBIEN REEMPLAZO $hijo_1()$ y $hijo_2()$

$$h(a) = h(b) \wedge completo(a) \wedge completo(b) \Rightarrow tam(bin(a,b,x)) \geq 2^{h(bin(a,b,x))} - 1$$

Asumimos que $h(a) = h(b) \wedge completo(a) \wedge completo(b)$, $\boxed{\text{Sino lo demo es trivial}}$

Y ADEMAS REEMPLAZO $t_1)$ y $h_1)$

vale, yo lo definiré al suponer $P(a) \wedge P(b)$.

$$1 + tam(a) + tam(b) \geq 2 \cdot 2^{max(h(a),h(b))} - 1$$

PERO, COMO VALE $P(a)$ y $P(b)$, y a y b son completos y $h(a) = h(b)$. Por HI

$$1 + tam(a) + tam(b) \geq 1 + 2^{h(a)} - 1 + 2^{h(b)} - 1 = 2 \cdot 2^{h(a)} - 1 = 2 \cdot 2^{max(h(a),h(b))} - 1$$ (hay que mencionarlo)

LUEGO, DEMOSTRAMOS ESTE PASO INDUCTIVO

Ahora queda el otro

$$(\forall a,b,c : abt(\alpha)) P(a) \wedge P(b) \wedge P(c) \Rightarrow (\forall x : \alpha) P(tern(a,b,c,x))$$

igual que antes, asumimos nada sobre a, b y c (PARA SACAR $\forall$), SUPONEMOS que $P(a) \wedge P(b) \wedge P(c)$ VALE (SI NO ES TRIVIAL) Y NO ASUMIMOS NADA SOBRE x, ENTONCES QUEDA

$$P(tern(a,b,c,x)) \Leftrightarrow completo(tern(a,b,c,x)) \Rightarrow tam(tern(a,b,c,x)) \geq 2^{h(tern(a,b,c,x))} - 1$$

APLICAMOS c), como $nil?(tern(a,b,c,x)) \equiv falso$ y lo sigue un $\vee_L$ ~~QUEDA~~ Y, APLICANDO $hijo_1$ y $hijo_2$, queda

$h(a) = h(b) \wedge completo(a) \wedge completo(b) \wedge (tern?(tern(a,b,c,x)) \Rightarrow_L h(b) = h(hijo_3(tern(a,b,c,x))) \wedge completo(hijo_3(tern(a,b,c,x)))) \Rightarrow tam(tern(a,b,c,x)) \geq 2^{h(tern(a,b,c,x))} - 1$

Como $tern?(tern(a,b,c,x)) \equiv true$ podemos deshacernos del $\Rightarrow_2$ y aplicar $hijo_3$

$$h(a)=h(b) \wedge completo(a) \wedge completo(b) \wedge h(b)=h(c) \wedge completo(c) \Rightarrow ta(tern(a,b,c,x)) \geq 2^{h(tern(a,b,c,x))} - 1$$

Asumimos que vale el antecedente, y aplicamos $t_2)$ y h_2

$$1+ta(a)+ta(b)+ta(c) \geq 2^{1+max(h(a),h(b),h(c))} - 1 = 2 \cdot 2^{max(h(a),h(b),h(c))} - 1$$

Como vale $P(a) \wedge P(b) \wedge P(c)$ y ~~...~~ el antecedente anterior se tiene que

$$1+ta(a)+ta(b)+ta(c) \overset{HI}{\geq} 1+2^{h(a)}-1+2^{h(b)}-1+2^{h(c)}-1 = 3 \cdot 2^{h(a)} - 2$$

$$= 2 \cdot 2^{h(a)} + 2^{h(a)} - 2 \geq 2 \cdot 2^{h(a)} + 2^{0} - 2 = 2 \cdot 2^{h(a)} - 1 = 2 \cdot 2^{max(h(a),h(b),h(c))} - 1$$

↳ por ser $h(a)$ un nat y 2^n creciente

Y entonces queda demostrado el segundo paso inductivo y, por lo tanto, que ~~a~~ $(\forall a : abt(\alpha))\ P(a)$ ✓

6

Orden 12

③ a) y b)

(I) LAS CLAVES DE cantUsos SON IGUALES A LA UNION DE LOS SIGNIFICADOS DE porPeligrosidad

$claves(e.cantUsos) = UnionCjtos(e.porPeligrosidad, claves(e.porPeligrosidad))$

(II) CADA REACTIVO TIENE UNA SOLA PELIGROSIDAD

$(\forall p,p':peligrosidad)\ def?(p, e.porPeligrosidad) \wedge def?(p', e.porPeligrosidad)$
$\Rightarrow_L (obtener(p, e.porPeligrosidad) \cap obtener(p', e.porPeligrosidad)) = \emptyset$

(III) NO HAY SIGNIFICADOS VACIOS en e.porPeligrosidad

$(\forall p:peligrosidad)\ def?(p, e.porPeligrosidad) \Rightarrow_L obtener(p, e.porPeligrosidad) \neq \emptyset$

(IV) NO HAY subProtocolos vacios

$(\forall n:nombre)\ def?(n, e.subProtocolos) \Rightarrow_L obtener(n, e.subProtocolos) \neq \langle\rangle$

(V) LAS CLAVES DE e.protocolos SON IGUALES A LA UNION DE LOS SIGNIFICADOS DE e.cantUsos

$claves(e.protocolos) = UnionCjtos(e.cantUsos, claves(e.cantUsos))$

(VI) LA UNION DE LOS SIGNIFICADOS DE e.protocolos ES IGUAL A LAS CLAVES DE e.subProtocolos

$claves(e.subProtocolos) = UnionSecu(e.protocolos, claves(e.protocolos))$

(VII) LA UNION DE LOS SIGNIFICADOS DE e.subProtocolos ES IGUAL A LAS CLAVES DE e.cantUsos

$claves(e.cantUsos) = UnionSecu(e.subProtocolos, claves(e.subProtocolos))$

~~(VIII) e.cantUsos, e.Protocolos y e.subProtocolos tienen sent. entre si~~

~~$(\forall r:reactivo)\ def?(r, e.cantUsos) \Rightarrow_L (\forall ...)\ obtener(r, e.cantUsos$~~

VIII e.cantUsos tiene sentido con e.protocolos y e.subProtocolos

$(\forall r:\text{reactivo})\ \text{def?}(r, e.\text{cantUsos}) \Rightarrow_L (\forall i:\text{id})\ i \in \text{obtener}(r, e.\text{cantUsos}) \Rightarrow_L$

$(\exists n:\text{nombre})\ \text{esta?}(n, \text{obtener}(i, e.\text{protocolos})) \wedge_L \text{esta?}(r, \text{obtener}(n, e.\text{subProtocolos}))$

~~$\wedge (\forall i:\text{id})\ \text{def?}(i, e.\text{protocolos}) \Rightarrow (\forall n:\text{nombre})\ \text{esta?}(n, \text{obtener}(i, e.\text{protocolos})) \Rightarrow_L (\forall r:\text{reactivo})$~~

~~IX LA SUMA DE PELIGROSIDADES DE NINGUN SUBPROTOCOLO SUPERA 100~~

~~$(\forall n:\text{nombre})\ \text{def?}(n, e.\text{subProtocolos}) \Rightarrow_L \text{sumaPel}(e.\text{porPeligrosidad}, \text{obtener}(n, e.\text{subProtocolos})) \leq 100$~~

REESCRIBO EN LA ULTIMA HOJA *

X e.protocolos y e.subProtocolos tienen sentido con e.cantUsos

$(\forall i:\text{id})\ \text{def?}(i, e.\text{protocolos}) \Rightarrow_L (\forall n:\text{nombre})\ \text{esta?}(n, \text{obtener}(i, e.\text{protocolos})) \Rightarrow_L$

$(\forall r:\text{reactivo})\ \text{esta?}(r, \text{obtener}(n, e.\text{subProtocolos})) \Rightarrow i \in \text{obtener}(r, e.\text{cantUsos})$

~~XI NO HAY SIGNIFICADOS VACIOS EN e.protocolos~~

~~$(\forall i:\text{id})\ \text{def?}(i, e.\text{protocolos}) \Rightarrow_L \text{obtener}(i, e.\text{protocolos}) \neq \langle\rangle$~~

XI Los id se numeran a partir del 1

$(\forall i:\text{id})(i \in \text{claves}(e.\text{protocolos}) \Rightarrow i \geq 1) \wedge \max(\text{claves}(e.\text{protocolos})) = \#(\text{claves}(e.\text{protocolos}))$

REP: estr $\longrightarrow$ boolean

$\text{REP}(e) \equiv \text{I} \wedge \text{II} \wedge \text{III} \wedge \text{IV} \wedge \text{V} \wedge \text{VI} \wedge \text{VII} \wedge_L \text{VIII} \wedge \cancel{\text{IX}} \wedge \text{X} \wedge \text{XI}$

Además, defino las siguientes operaciones auxiliares

UnionCjtos: $\text{dicc}(\alpha, \text{conj}(\beta))\ d \times \text{conj}(\alpha)\ c \longrightarrow \text{conj}(\beta)$
$\{c \subseteq \text{claves}(d)\}$

UnionSecu: $\text{dicc}(\alpha, \text{secu}(\beta))\ d \times \text{conj}(\alpha)\ c \longrightarrow \text{conj}(\beta)$
$\{c \subseteq \text{claves}(d)\}$

Conjuntizar: $\text{secu}(\alpha) \longrightarrow \text{conj}(\alpha)$

Peligrosidad: $\text{dicc}(\text{peligrosidad}, \text{cjto}(\alpha))\ d \times \alpha\ a \times \text{conj}(\text{peligrosidad})\ c \longrightarrow \text{peligrosidad}$
$\{(\exists!\, p:\text{peligrosidad})(\text{def?}(p, d) \wedge_L a \in \text{obtener}(p, d)) \wedge c \subseteq \text{claves}(d)\}$

SumaPel : dicc(peligrosidad, cjto(α))^d × sec(α)^s → ~~nat~~ peligrosidad
{UnionCjto(d, claves(d)) ⊇ Conjuntizar(s)}

UnionCjtos(d, c) ≡ if ∅?(c) then ∅ else
obtener(dameUno(c), d) ∪ UnionCjtos(d, sinUno(c)) fi

UnionSecu(d, c) ≡ if ∅?(c) then ∅ else
conjuntizar(obtener(dameUno(c), d)) ∪
UnionSecu(d, sinUno(c)) fi

Conjuntizar(s) ≡ if s = <> then ∅ else
Ag(prim(s), Conjuntizar(fin(s))) fi

Peligrosidad(d, a, c) ≡ ~~if ∅?(c) then~~ if a ∈ obtener(dameUno(c), d) then
dameUno(c) else
Peligrosidad(d, a, sinUno(c)) fi

SumaPoli(d, s) ≡ if s = <> then 0 else
Peligrosidad(d, prim(s), claves(d)) + SumaPoli(d, fin(s))
fi

C. Abs : estr e → Experimentos {REP(e)}

Abs(e) ≡ ex : Experimento | cantExperimentos(ex) = #(claves(e.protocolos)) ∧
(∀i : id) i ∈ claves(e.protocolos) ⇒$_L$ verExperimento(ex, i) =
concatenar(obtener(i, e.protocolos))

~~#(claves(e.peligrosidad))~~.
Perdón, no vi que estaba atrás.

$\text{concatenar} : \text{secu(nombre)} \times \text{estr} \rightarrow \text{secu(tupla(reactivo, peligrosidad))}$

$\text{protocolizar} : \text{nombre} \times \text{estr} \rightarrow \text{secu(tupla(reactivo, peligrosidad))}$

$\text{concatenar}(s, e) \equiv \text{if } s = \langle\rangle \text{ then } \langle\rangle \text{ else}$

$\quad \text{protocolizar}(\text{prim}(s), e) \bullet \text{concatenar}(\text{fin}(s), e) \text{ fi}$

↳ no encuentro la def. No importa.

(*) (1X) La suma de peligrosidades de ningún protocolo supera 100

$(\forall i : id)\ \text{def?}(i, e.\text{protocolos}) \Rightarrow_L \text{sumaPeliNom}(\text{obtener}(i, e.\text{protocolos}), e) \leq 100$

$\text{sumaPeliNom} : \text{secu}(\alpha)^s \times \text{estr}^e \rightarrow \text{peligrosidad}$

$\{s \subseteq \text{claves}(e.\text{subprotocolos})\}$

$\text{sumaPeliNom}(s, e) \equiv \text{if } s = \langle\rangle \text{ then } 0 \text{ else}$

$\quad \text{sumaPeli}(e.\text{porPeligrosidad}, \text{obtener}(\text{prim}(s), e.\text{subProtocolos})) +$

$\quad \text{sumaPeliNom}(\text{fin}(s), e) \text{ fi}$